

Unbiased degree-preserving randomisation of directed binary networks

ES Roberts^{†‡} and ACC Coolen^{†‡§}

[†] Department of Mathematics, King's College London, The Strand, London WC2R 2LS, United Kingdom

[‡] Randall Division of Cell and Molecular Biophysics, King's College London,

New Hunts House, London SE1 1UL, United Kingdom and

[§] London Institute for Mathematical Sciences, 35a South St, Mayfair, London W1K 2XF, United Kingdom

Randomising networks using a naive ‘accept-all’ edge-swap algorithm is generally biased. Building on recent results for nondirected graphs, we construct an ergodic detailed balance Markov chain with non-trivial acceptance probabilities for directed graphs, which converges to a strictly uniform measure and is based on edge swaps that conserve all in- and out-degrees. The acceptance probabilities can also be generalized to define Markov chains that target any alternative desired measure on the space of directed graphs, in order to generate graphs with more sophisticated topological features. This is demonstrated by defining a process tailored to the production of directed graphs with specified degree-degree correlation functions. The theory is implemented numerically and tested on synthetic and biological network examples.

I. INTRODUCTION

When seeking to assess the statistical relevance of observations made in real networks, there are three routes available. One could generate null-model networks for hypothesis testing from scratch, constrained by the values of observed parameters in the real network (e.g. using the Molloy-Reed stub joining method [1], or the Barabási-Albert preferential attachment model [2]). Alternatively, one could generate null-model networks by randomising the original network, using dynamical rules that leave the values of relevant parameters invariant [3]. The final option is to use analytical methods to find ensemble averages for the observable of interest, see e.g. [4, 5].

The null-model approach is appealing in its conceptual simplicity. It effectively provides synthetic ‘data’, which can be analysed in the same way as the real dataset. One can then learn which observed properties are particular to the real dataset, and which are common within the ensemble.

Applications of network null-models are wide ranging and central to network science. [6] applies null models to identify over-represented ‘motifs’ in the transcriptional regulation network of *E. coli*. [7] discusses adapting the Watts-Strogatz method to generating random networks to model power grids. [8] explores motifs found within an interfirm network. [9] uses network null-models to study social networks. [10] compares topological properties of interaction and transcription regulatory networks in yeast with randomised ‘null model’ networks and postulated that links between highly connected proteins are suppressed in protein interaction networks. [11] discusses the challenges of specifying a suitable matrix null model in the field of ecology.

It is crucial that the synthetic networks generated as null models are representative of the underlying ensembles. Any inadvertent bias in the network generation process may invalidate the hypothesis test. It is therefore worrying that the two most popular methods to randomise or generate null networks are in fact known to be biased. The common implementation of the stub-joining method, where invalid edges are rejected but the process subsequently continues (as opposed to starting from the beginning of the whole process), is known to be biased [12–14]; in fact, even if upon invalid edge rejection

the stub-joining process is restarted, it is not clear whether the graphs produced would be unbiased (we are not aware of any published proof). Similarly, the conventional ‘accept-all’ edge swap process, see e.g. [15], is also known to be biased [16]: graphs on which many swaps can be executed are generated more often. The effects of these biases may in the past not always have been serious [17], but using biased algorithms for producing null models is fundamentally unsound, and unacceptable when there are rigorous unbiased alternatives [16].

In this paper we build on the work of [16] and [3] and define a Markov Chain Monte Carlo process, based on ergodic in- and out-degree preserving edge-swap moves that act on *directed* networks. We first calculate correct move acceptance probabilities for the process to sample the space of all allowed directed graphs uniformly. We then extend the theory in order for the process to evolve to any desired measure on the space of directed graphs. Attention is paid to adapting our results for efficient numerical implementation. We also identify under which circumstances the error made by doing ‘accept all’ edge swaps is immaterial. We apply our theory to real and synthetic networks.

II. AN ERGODIC AND UNBIASED RANDOMISATION PROCESS WHICH PRESERVES IN- AND OUT-DEGREES

A. Edge swap moves

The canonical moves for degree-preserving randomisation of graphs are the so-called ‘edge swaps’, see e.g. [16, 18, 19]. The undirected version of the edge swap is illustrated in figure 1; a generalisation to directed graphs is found in [3]. The authors of [3] define a move - which we will refer to as a *square swap* - starting from a set of four entries from the connectivity matrix $c \in \{0, 1\}^{N^2}$ of a directed binary N -node graph, defined by node pairs $\{(i_1, j_1), (i_1, j_2), (i_2, j_2), (i_2, j_1)\}$ such that the corresponding entries $\{c_{i_1 j_1}, c_{i_1, j_2}, c_{i_2, j_2}, c_{i_2, j_1}\}$ are alternately 0 and 1, and not ‘structural’ (i.e. they are allowed to vary). As for the undirected case, the elementary edge swap move is defined by swapping the 0 and 1 entries, i.e. $\{c_{i_1 j_1}, c_{i_1, j_2}, c_{i_2, j_2}, c_{i_2, j_1}\} \rightarrow \{1 - c_{i_1 j_1}, 1 - c_{i_1, j_2}, 1 - c_{i_2, j_2}, 1 - c_{i_2, j_1}\}$.

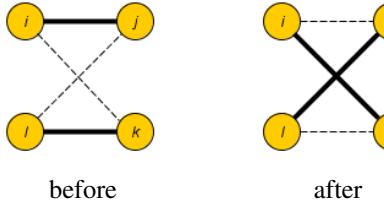


FIG. 1. The undirected edge swap. This is the canonical choice for the elementary moves of an ergodic degree-preserving randomisation process on undirected networks.

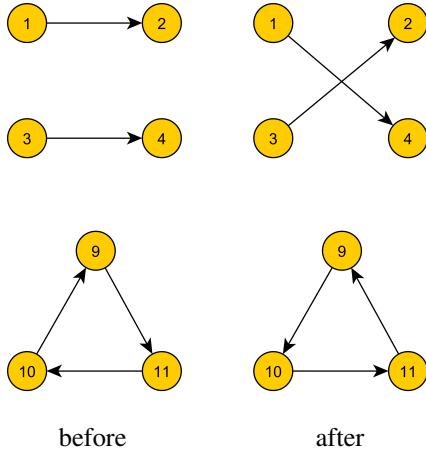


FIG. 2. The *square swap* (top) and *triangle swap* (bottom). In combination these two represent the canonical choice for the elementary moves of an ergodic degree-preserving randomisation process on directed networks without self-interactions.

The authors of [3] prove that, if self interactions are permitted, repeated application of such moves can transform any binary matrix \mathbf{c}_A to any other binary matrix \mathbf{c}_B with the same in- and out- degree distributions. However, if we require in addition that the diagonal elements of all \mathbf{c} are 0 (i.e. we forbid self-interactions), then the edge swap defined above is no longer sufficient to ensure ergodicity. To remedy this problem the authors of [3] introduce a further move, which we will call a *triangle swap*. This move also gives us the simplest demonstration of two valid configurations that cannot be connected by square-type swaps. The *square swap* and the *triangle swap* are illustrated in figure 2; in combination these two moves allow us to transform between any two directed binary matrices which have the same in- and out-degrees, even if self-interactions are forbidden [3].

A stochastic process defined as accepting all randomly selected moves from the above set is ergodic but biased. This was already observed in [3], where the authors proposed a ‘Switch & Hold’ algorithm, which involves the number of states accessible in one move from a configuration (its mobility), and the maximum possible number of states accessible in one move from any network in the ensemble (the degrees of a hyper-graph, in the language of later publications). In [16] the problem was studied for undirected graphs; it was shown

how move acceptance probabilities should be defined to guarantee stochastic evolution by edge swapping to any desired measure on the space of nondirected graphs. The analysis in [16] is quite general, and briefly reproduced in section II B below. Here we will adapt their calculations to directed graphs and include the new moves defined by [3]. This will result in a Markovian process based on edge swapping that will equilibrate to any desired measure on the space of *directed* graphs.

B. Outline of the general theory

This section briefly summarizes results of [16] which will be used in the next section. We define an adjacency matrix $\mathbf{c} = \{c_{ij}\}$, where $c_{ij} = 1$ if and only if there is a directed link from node j to node i . We denote the set of all such graphs as \mathcal{C} . The aim is to define and study constrained Markov chains for the evolution of \mathbf{c} in some subspace $\Omega \in \mathcal{C}$. This is a discrete time stochastic process, where the probability $p_t(\mathbf{c})$ of observing a graph \mathbf{c} at time t evolves according to

$$\forall \mathbf{c} \in \Omega : p_{t+1}(\mathbf{c}) = \sum_{\mathbf{c}' \in \Omega} W(\mathbf{c}|\mathbf{c}') p_t(\mathbf{c}') \quad (1)$$

where $t \in \mathbb{N}$ and $W(\mathbf{c}|\mathbf{c}')$ is a transition probability. We require the process to have three additional properties:

1. Each elementary move F can only act on a subset of all possible graphs.
2. The process converges to the invariant measure

$$p_\infty(\mathbf{c}) = Z^{-1} e^{-H(\mathbf{c})}$$

3. Each move F has a unique inverse, which acts on the same subset of states as F itself.

We exclude the identity move from the set Φ of all moves, and we define an indicator function $I_F(\mathbf{c})$ where $I_F(\mathbf{c}) = 1$ iff the move $\mathbf{c} \rightarrow F\mathbf{c}$ is allowed. The transition probabilities are constructed to obey detailed balance

$$\forall \mathbf{c}, \mathbf{c}' \in \Omega : W(\mathbf{c}|\mathbf{c}') p_\infty(\mathbf{c}') = W(\mathbf{c}'|\mathbf{c}) p_\infty(\mathbf{c}) \quad (2)$$

At each step a candidate move $F \in \Phi$ is drawn with probability $q(F|\mathbf{c}')$, where \mathbf{c}' is the current state. The move is accepted with some probability $A(F\mathbf{c}'|\mathbf{c}')$. In combination this leads to

$$W(\mathbf{c}|\mathbf{c}') = \sum_{F \in \Phi} q(F|\mathbf{c}') [\delta_{\mathbf{c}, F\mathbf{c}'} A(F\mathbf{c}'|\mathbf{c}') + \delta_{\mathbf{c}', \mathbf{c}} [1 - A(F\mathbf{c}'|\mathbf{c}')]] \quad (3)$$

Insertion into (2) then leads to the following conditions which must be satisfied by $A(F\mathbf{c}'|\mathbf{c}')$ and $q(F|\mathbf{c}')$:

$$(\forall \mathbf{c} \in \Omega)(\forall F \in \Phi) : \quad (4)$$

$$q(F|\mathbf{c}) A(F\mathbf{c}|\mathbf{c}) e^{-H(\mathbf{c})} = q(F^{-1}|\mathbf{c}) A(\mathbf{c}|F\mathbf{c}) e^{-H(F\mathbf{c})}$$

We define the mobility $n(\mathbf{c})$ to be the number of moves which can act on each state: $n(\mathbf{c}) = \sum_{F \in \Phi} I_F(\mathbf{c})$. If the candidate

moves are drawn randomly with equal probabilities from the set of all moves allowed to act, we find (4) reducing to

$$A(\mathbf{c}|\mathbf{c}') = \frac{n(\mathbf{c}')e^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]} }{n(\mathbf{c}')e^{-\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]} + n(\mathbf{c})e^{\frac{1}{2}[H(\mathbf{c})-H(\mathbf{c}')]} } \quad (5)$$

If we make the simplest choice $H(\mathbf{c}) = \text{const}$, the above process will asymptotically sample all graphs with the imposed degree sequence uniformly. To sample this constrained space of graphs with alternative nontrivial probabilities $p_\infty(\mathbf{c})$ we would choose $H(\mathbf{c}) = -\log p_\infty(\mathbf{c}) + \text{const}$.

Equation 4 also shows what would happen if we were to sample with $A(\mathbf{c}|\mathbf{c}') \equiv 1$ for all $(\mathbf{c}, \mathbf{c}')$, i.e. for ‘accept all’ edge swapping: the detailed balance condition would give

$$(\forall \mathbf{c} \in \Omega)(\forall F \in \Phi) : e^{-H(\mathbf{c})}n(\mathbf{c}) = e^{-H(F\mathbf{c})}n(F\mathbf{c}) \quad (6)$$

For this to be satisfied we require both sides of the expression to evaluate to a constant. Hence $e^{-H(\mathbf{c})} \propto n(\mathbf{c})$, so the naive process will converge to the non-uniform measure

$$p_\infty(\mathbf{c}) = Z^{-1}n(\mathbf{c}) \quad (7)$$

This is the undesirable bias of ‘accept-all’ edge-swapping. It has a clear intuitive explanation. The mobility $n(\mathbf{c})$ is the number of allowed moves on network \mathbf{c} , which is equal to the number of inverse moves through which \mathbf{c} can be reached in one step from another member of the ensemble. The likelihood of seeing a network \mathbf{c} upon equilibration of the process is proportional to the number of entry points that \mathbf{c} offers the process.

C. Calculation of the mobilities for directed networks

Since the two types of moves required for ergodic evolution of directed graphs, viz. the square swap and the triangle swap, are clearly distinct, the mobility of a graph \mathbf{c} is given by $n(\mathbf{c}) = n_\square(\mathbf{c}) + n_\Delta(\mathbf{c})$, where $n_\square(\mathbf{c})$ and $n_\Delta(\mathbf{c})$ count the number of possible moves of each type that can be executed on \mathbf{c} .

To find $n_\square(\mathbf{c})$ we need to calculate how many distinct link-alternating cycles of length 4 can be chosen in graph \mathbf{c} . We exclude self-interactions, so our cycles must involve 4 distinct nodes. The total number of such moves can be written as

$$n_\square(\mathbf{c}) = \frac{1}{2} \sum_{ijkl} \bar{\delta}_{jk} \bar{\delta}_{li} \bar{\delta}_{ik} \bar{\delta}_{jl} c_{ij} c_{kl} \bar{c}_{kj} \bar{c}_{il} \quad (8)$$

where the pre-factor compensates for the symmetry, and where we used the short-hands $\bar{c}_{kj} = 1 - c_{kj}$ and $\bar{\delta}_{jk} = 1 - \delta_{jk}$. Expanding these shorthands in (8) gives after some further bookkeeping of terms, and with $(\mathbf{c}^\dagger)_{ij} = c_{ji}$:

$$\begin{aligned} n_\square(\mathbf{c}) &= \frac{1}{2} \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}\mathbf{c}^\dagger) - \sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} + \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}) + \frac{1}{2} \text{Tr}(\mathbf{c}^2) \\ &\quad + \frac{1}{2} N^2 \langle k \rangle^2 - \sum_j k_j^{\text{in}} k_j^{\text{out}} \end{aligned} \quad (9)$$

with $\langle k \rangle = N^{-1} \sum_i k_i^{\text{in}} = N^{-1} \sum_i k_i^{\text{out}}$. We next repeat the calculation for the case of the *triangle swap*. For easier manipu-

lations, we introduce a new matrix \mathbf{c}^\dagger of double links, defined via $(\mathbf{c}^\dagger)_{ij} = c_{ij}c_{ji}$. We then find

$$\begin{aligned} n_\Delta(\mathbf{c}) &= \frac{1}{3} \sum_{ijk} \bar{\delta}_{ij} \bar{\delta}_{jk} \bar{\delta}_{ki} c_{ij} c_{jk} c_{ki} \bar{c}_{ji} \bar{c}_{kj} \bar{c}_{ik} \\ &= \frac{1}{3} \{ \text{Tr}(\mathbf{c}^3) - 3\text{Tr}(\mathbf{c}^\dagger \mathbf{c}^2) + 3\text{Tr}(\mathbf{c}^{\dagger 2} \mathbf{c}) + -\text{Tr}(\mathbf{c}^{\dagger 3}) \} \\ &= \frac{1}{3} \text{Tr}((\mathbf{c} - \mathbf{c}^\dagger)^3) \end{aligned} \quad (10)$$

In combination, (9) and (10) give us an explicit and exact formula for the graph mobility $n(\mathbf{c}) = n_\square(\mathbf{c}) + n_\Delta(\mathbf{c})$, and hence via (5) a fully exact MCMC process for generating random graphs with prescribed sequences and any desired probability measure in the standard form $Z^{-1} \exp[-H(\mathbf{c})]$. Since (9,10) cannot be written in terms of the degree sequence only, neglecting the mobility (as with accept-all edge swapping) would always introduce a bias into the sampling process.

III. PROPERTIES AND IMPACT OF GRAPH MOBILITY

A. Bounds on the mobility

We will now derive bounds on the sizes of the mobility terms. This may show for which types of networks the application of ‘accept all’ edge swapping (which ignores the mobility terms) is most dangerous, and for which networks the unwanted bias may be small. We first observe that

$$n_\Delta(\mathbf{c}) = \frac{1}{3} \sum_{ijk} c_{ij}(1 - c_{ji})c_{jk}(1 - c_{kj})c_{ki}(1 - c_{ik}) \leq \frac{1}{3} \text{Tr}(\mathbf{c}^3)$$

Hence, the mobility $n(\mathbf{c}) = n_\square(\mathbf{c}) + n_\Delta(\mathbf{c})$ obeys

$$\begin{aligned} n(\mathbf{c}) &\leq \frac{1}{2} \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}\mathbf{c}^\dagger) - \sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} + \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}) + \frac{1}{2} \text{Tr}(\mathbf{c}^2) \\ &\quad + \frac{1}{2} N^2 \langle k \rangle^2 - \sum_j k_j^{\text{in}} k_j^{\text{out}} + \frac{1}{3} \text{Tr}(\mathbf{c}^3) \end{aligned} \quad (11)$$

We find upper bounds for most of the terms above by applying the simple inequality $c_{ij}c_{kl} \leq \frac{1}{2}(c_{ij} + c_{kl})$, which gives e.g.

$$\text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}) \leq \frac{N}{2} [\langle k^{\text{in}} \rangle^2 + \langle k^{\text{out}} \rangle^2] \quad \text{Tr}(\mathbf{c}^2) \leq N \langle k \rangle \quad (12)$$

$$\text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}\mathbf{c}^\dagger) \leq \sum_{ij} k_i^{\text{in}} c_{ij} k_i^{\text{out}} \quad \text{Tr}(\mathbf{c}^3) \leq N \langle k^{\text{in}} k^{\text{out}} \rangle \quad (13)$$

An upper bound on $\sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}}$ follows from the observation that if $c_{ij} = 1$ then certainly $k_i^{\text{in}} \geq 1$ and $k_j^{\text{out}} \geq 1$. Hence

$$\begin{aligned} \sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} &\geq \frac{1}{2} \sum_{ij} [c_{ij} k_j^{\text{out}} + k_i^{\text{in}} c_{ij}] \\ &= \frac{1}{2} N [\langle k^{\text{in}} \rangle^2 + \langle k^{\text{out}} \rangle^2] \end{aligned} \quad (14)$$

Combining (12,13,14) with (11) then gives

$$n(\mathbf{c}) \leq \frac{N}{2} \left[N\langle k \rangle^2 + \langle k \rangle + \frac{1}{2} [\langle k^{\text{in}} \rangle^2 + \langle k^{\text{out}} \rangle^2] - \frac{4}{3} \langle k^{\text{in}} k^{\text{out}} \rangle \right] \quad (15)$$

Next we calculate a lower bound for $n(\mathbf{c})$. We use simple identities such as

$$\text{Tr}(\mathbf{c}^2) \geq 0 \quad n_{\Delta}(\mathbf{c}) \geq 0 \quad \text{Tr}(\mathbf{c}\mathbf{c}^{\dagger}\mathbf{c}) \geq 0 \quad (16)$$

and

$$\begin{aligned} \text{Tr}(\mathbf{c}\mathbf{c}^{\dagger}\mathbf{c}\mathbf{c}^{\dagger}) &\geq \frac{1}{2} \sum_{ijk\ell} c_{ji} c_{jk} c_{\ell k} c_{\ell i} (\delta_{j\ell} + \delta_{ik}) \\ &= N[\langle k^{\text{in}} \rangle^2 + \langle k^{\text{out}} \rangle^2] \end{aligned} \quad (17)$$

We now find

$$\begin{aligned} n(\mathbf{c}) &\geq \frac{1}{2} N[\langle k^{\text{in}} \rangle^2 + \langle k^{\text{out}} \rangle^2] + \frac{1}{2} N^2 \langle k \rangle^2 - \sum_j k_j^{\text{in}} k_j^{\text{out}} \\ &\quad - \sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} \end{aligned} \quad (18)$$

We finally need an upper bound for $\sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}}$, which we write in terms of $k_{\max}^{\text{in}} = \max_i k_i^{\text{in}}$ and $k_{\max}^{\text{out}} = \max_i k_i^{\text{out}}$:

$$\begin{aligned} \sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} &\leq \frac{1}{2} \sum_{ij} [k_{\max}^{\text{in}} c_{ij} k_j^{\text{out}} + k_i^{\text{in}} c_{ij} k_{\max}^{\text{out}}] \\ &= \frac{1}{2} N [k_{\max}^{\text{in}} \langle k^{\text{out}} \rangle^2 + k_{\max}^{\text{out}} \langle k^{\text{in}} \rangle^2] \end{aligned} \quad (19)$$

We thus obtain our lower bound for the mobility:

$$n(\mathbf{c}) \geq \frac{N}{2} \left[N\langle k \rangle^2 + \langle (k^{\text{in}} - k^{\text{out}})^2 \rangle - k_{\max}^{\text{in}} \langle k^{\text{out}} \rangle^2 - k_{\max}^{\text{out}} \langle k^{\text{in}} \rangle^2 \right] \quad (20)$$

B. Identification of graph types most likely to be biased by ‘accept all’ edge swapping

We know from (5) that unbiased sampling of graphs, i.e. $p(\mathbf{c}) = 1/|\Omega|$ for all $\mathbf{c} \in \Omega$, requires using the following state-dependent acceptance probabilities in the edge swap process:

$$A(\mathbf{c}|\mathbf{c}') = [1 + n(\mathbf{c})/n(\mathbf{c}')]^{-1} \quad (21)$$

We now investigate under which conditions one will in large graphs effectively find $n(\mathbf{c})/n(\mathbf{c}') \rightarrow 1$ for all $\mathbf{c}, \mathbf{c}' \in \Omega$, so that the sampling bias would be immaterial. Let us define

$$\Delta n = \max_{\mathbf{c}, \mathbf{c}' \in \Omega} |n(\mathbf{c}) - n(\mathbf{c}')| = \max_{\mathbf{c} \in \Omega} n(\mathbf{c}) - \min_{\mathbf{c} \in \Omega} n(\mathbf{c}) \quad (22)$$

Using the two bounds (15,20) we immediately obtain

$$\begin{aligned} \Delta n &\leq \frac{N}{2} \left[\langle k \rangle - \frac{1}{2} [\langle k^{\text{in}} \rangle^2 + \langle k^{\text{out}} \rangle^2] + \frac{2}{3} \langle k^{\text{in}} k^{\text{out}} \rangle \right. \\ &\quad \left. + k_{\max}^{\text{in}} \langle k^{\text{out}} \rangle^2 + k_{\max}^{\text{out}} \langle k^{\text{in}} \rangle^2 \right] \\ &= \frac{N}{2} \left[\langle k \rangle - \frac{1}{6} [\langle k^{\text{in}} \rangle^2 + \langle k^{\text{out}} \rangle^2] - \frac{1}{3} \langle (k^{\text{in}} - k^{\text{out}})^2 \rangle \right. \\ &\quad \left. + k_{\max}^{\text{in}} \langle k^{\text{out}} \rangle^2 + k_{\max}^{\text{out}} \langle k^{\text{in}} \rangle^2 \right] \\ &\leq \frac{N}{2} \left[\langle k \rangle + k_{\max}^{\text{in}} \langle k^{\text{out}} \rangle^2 + k_{\max}^{\text{out}} \langle k^{\text{in}} \rangle^2 \right] \end{aligned} \quad (23)$$

Clearly $1 - \Delta n/n(\mathbf{c}) \leq n(\mathbf{c}')/n(\mathbf{c}) \leq 1 + \Delta n/n(\mathbf{c})$, so in view of (21) we are interested in the ratio $\Delta n/n(\mathbf{c})$, for which we find

$$\frac{\Delta n}{n(\mathbf{c})} \leq \frac{\langle k \rangle + k_{\max}^{\text{in}} \langle k^{\text{out}} \rangle^2 + k_{\max}^{\text{out}} \langle k^{\text{in}} \rangle^2}{N\langle k \rangle^2 - k_{\max}^{\text{in}} \langle k^{\text{out}} \rangle^2 - k_{\max}^{\text{out}} \langle k^{\text{in}} \rangle^2} \quad (24)$$

So we can be confident that the impact of the graph mobility on the correct acceptance probabilities (21) is immaterial if

$$\frac{1}{\langle k \rangle} + \frac{2}{\langle k \rangle^2} (k_{\max}^{\text{in}} \langle k^{\text{out}} \rangle^2 + k_{\max}^{\text{out}} \langle k^{\text{in}} \rangle^2) \ll N \quad (25)$$

We see from this that we can apply the ‘accept all’ edge-swap process with confidence when we are working with a large network with a narrow degree distribution.

IV. MOBILITIES OF SIMPLE GRAPH EXAMPLES

In this section we confirm the validity of the mobility formulae (9,10) for several simple examples of directed graphs.

1. Two isolated bonds:

Here we have $c_{12} = 1$, $c_{34} = 1$, and $c_{ij} = 0$ for all $(i, j) \notin \{(1, 2), (3, 4)\}$. It is immediately clear that $\mathbf{c}^{\dagger} = \mathbf{0}$, and

$$\begin{aligned} \sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} &= 2, \quad \sum_j k_j^{\text{out}} k_j^{\text{in}} = 0, \quad \langle k \rangle = \frac{2}{N} \\ \text{Tr}(\mathbf{c}^2) = \text{Tr}(\mathbf{c}^3) = \text{Tr}(\mathbf{c}\mathbf{c}^{\dagger}\mathbf{c}) &= 0, \quad \text{Tr}(\mathbf{c}\mathbf{c}^{\dagger}\mathbf{c}\mathbf{c}^{\dagger}) = 2 \end{aligned}$$

Insertion into (9,10) gives $n_{\square}(\mathbf{c}) = 1$ and $n_{\Delta}(\mathbf{c}) = 0$. As we would expect: only one (square) move is permitted.

2. Isolated triangle:

This example is defined by $c_{12} = c_{23} = c_{31} = 1$, with $c_{ij} = 0$ for all $(i, j) \notin \{(1, 2), (2, 3), (3, 1)\}$. Again we have $\mathbf{c}^{\dagger} = \mathbf{0}$, but now

$$\begin{aligned} \sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} &= 3, \quad \sum_j k_j^{\text{out}} k_j^{\text{in}} = 3, \quad \langle k \rangle = \frac{3}{N} \\ \text{Tr}(\mathbf{c}^2) = \text{Tr}(\mathbf{c}\mathbf{c}^{\dagger}\mathbf{c}) &= 0, \quad \text{Tr}(\mathbf{c}^3) = 3, \quad \text{Tr}(\mathbf{c}\mathbf{c}^{\dagger}\mathbf{c}\mathbf{c}^{\dagger}) = 3 \end{aligned}$$

This results in $n_{\square}(\mathbf{c}) = 0$ and $n_{\Delta}(\mathbf{c}) = 1$. The only possible move is reversal of the directed triangle.

3. Complete (fully connected) graph:

Here $c_{ij} = 1 - \delta_{ij}$, and no edge swaps are possible. All nodes have $k_i^{\text{in}} = k_i^{\text{out}} = N-1$, and since $\mathbf{c}^\dagger = \mathbf{c}$ we know that $n_\Delta(\mathbf{c}) = 0$. This connectivity matrix, also featured in [16], has eigenvalues $\lambda = N-1$ (multiplicity 1) and $\lambda = -1$ (multiplicity $N-1$). Hence

$$\begin{aligned} \sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} &= N(N-1)^3, \quad \sum_j k_j^{\text{out}} k_j^{\text{in}} = N(N-1)^2 \\ \text{Tr}(\mathbf{c}^2) &= \sum_i \lambda_i^2 = N(N-1), \\ \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}) &= \text{Tr}(\mathbf{c}^3) = \sum_i \lambda_i^3 = N(N-1)(N-2), \\ \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}\mathbf{c}^\dagger) &= \text{Tr}(\mathbf{c}^4) = \sum_i \lambda_i^4 = (N-1)[(N-1)^3 + 1] \end{aligned}$$

Assembling the entire expression for the square mobility term (9) indeed gives the correct value $n_\square(\mathbf{c}) = 0$.

4. Directed spanning ring:

This directed graph, defined by $c_{ij} = \delta_{i+1,j}$ modulo N , gives a ring with a flow around it. We choose $N > 2$. Once more $\mathbf{c}^\dagger = \mathbf{0}$, and we obtain for the relevant terms

$$\begin{aligned} \sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} &= \sum_j k_j^{\text{out}} k_j^{\text{in}} = N, \quad \langle k \rangle = 1 \\ \text{Tr}(\mathbf{c}^2) = \text{Tr}(\mathbf{c}^3) &= \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}) = 0, \quad \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}\mathbf{c}^\dagger) = N \end{aligned}$$

The final result, $n_\square(\mathbf{c}) = \frac{1}{2}N(N-3)$ and $n_\Delta(\mathbf{c}) = 0$, is again what we would expect. As soon as one first bond to participate in an edge swap is picked (for which there are N options), there are $N-3$ possibilities for the second (since the already picked bond and its neighbours are forbidden). The factor 2 corrects for over-counting.

5. Bidirectional spanning ring:

Our final example is the nondirected version of the previous graph, viz. $c_{ij} = \delta_{i,j-1} + \delta_{i,j+1}$ modulo N , with $N > 2$. Since $\mathbf{c}^\dagger = \mathbf{c}$ we have $n_\Delta(\mathbf{c}) = 0$. Now

$$\begin{aligned} \sum_{ij} k_i^{\text{in}} c_{ij} k_j^{\text{out}} &= 8N, \quad \sum_j k_j^{\text{out}} k_j^{\text{in}} = 4N, \quad \langle k \rangle = 2 \\ \text{Tr}(\mathbf{c}^2) &= 2N, \quad \text{Tr}(\mathbf{c}^3) = \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}) = 0 \\ \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}\mathbf{c}^\dagger) &= 6N \end{aligned}$$

We thereby find $n_\square(\mathbf{c}) = 2N(N-4)$. This is double the mobility evaluated in [16], since every move in the undirected version of the network corresponds to two possible moves in the directed version of the network.

V. A RANDOMISATION PROCESS WHICH PRESERVES DEGREES AND TARGETS DEGREE-DEGREE CORRELATIONS

So far we applied formula (5) for the canonical acceptance probabilities for directed graph edge swapping to the prob-

lem of generating graphs with prescribed in- and out-degrees (\mathbf{k}^{in} , \mathbf{k}^{out}) and a uniform measure. Here consider how to generate graphs which, in addition, display certain degree correlations. We first rewrite (5) as

$$A(\mathbf{c}|\mathbf{c}') = \left[1 + \frac{n(\mathbf{c})}{n(\mathbf{c}')} e^{H(\mathbf{c})-H(\mathbf{c}')} \right]^{-1} \quad (26)$$

These probabilities (26) ensure the edge-swapping process evolves into the stationary state on $\Omega = \{\mathbf{c} \in \{0,1\}^{N^2} \mid \mathbf{k}^{\text{in}}(\mathbf{c}) = \mathbf{k}^{\text{in}}, \mathbf{k}^{\text{out}}(\mathbf{c}) = \mathbf{k}^{\text{out}}\}$ defined by $p_\infty(\mathbf{c}) = Z^{-1} \exp[-H(\mathbf{c})]$. The full degree-degree correlation structure of a directed graph \mathbf{c} is captured by the joint degree distribution of connected nodes

$$W(\vec{k}, \vec{k}'|\mathbf{c}) = \frac{1}{N\langle k \rangle} \sum_{ij} c_{ij} \delta_{\vec{k}_i, \vec{k}_i(\mathbf{c})} \delta_{\vec{k}'_j, \vec{k}_j(\mathbf{c})} \quad (27)$$

with $\vec{k} = (k^{\text{in}}, k^{\text{out}})$. The maximum entropy distribution on Ω , viz. all directed graphs with prescribed in- and out-degree sequences, that has the distribution (27) imposed as a soft constraint, i.e. $\sum_{\mathbf{c} \in \Omega} p(\mathbf{c}) W(\vec{k}, \vec{k}'|\mathbf{c}) = W(\vec{k}, \vec{k}')$ for all (\vec{k}, \vec{k}') , is

$$\begin{aligned} p(\mathbf{c}) &= Z^{-1} \prod_i \delta_{\vec{k}_i, \vec{k}_i(\mathbf{c})} \\ &\times \prod_{ij} \left[\frac{\langle k \rangle}{N} Q(\vec{k}_i, \vec{k}_j) \delta_{c_{ij}, 1} + \left(1 - \frac{\langle k \rangle}{N} Q(\vec{k}_i, \vec{k}_j)\right) \delta_{c_{ij}, 0} \right] \end{aligned} \quad (28)$$

(see [20]), in which $Q(\vec{k}, \vec{k}') = W(\vec{k}, \vec{k}')/p(\vec{k})p(\vec{k}')$ and $p(\vec{k}) = p(k^{\text{in}}, k^{\text{out}})$. It is now trivial, following [16], to ensure that our MCMC process evolves to the measure (28) by choosing $H(\mathbf{c}) = -\log p(\mathbf{c})$ in the probabilities (26). This gives

$$\begin{aligned} A(\mathbf{c}|\mathbf{c}') &= \left[1 + \frac{n(\mathbf{c})}{n(\mathbf{c}')} \prod_{ij} \frac{\frac{\langle k \rangle}{N} Q(\vec{k}_i, \vec{k}_j) c'_{ij} + \left(1 - \frac{\langle k \rangle}{N} Q(\vec{k}_i, \vec{k}_j)\right) (1 - c'_{ij})}{\frac{\langle k \rangle}{N} Q(\vec{k}_i, \vec{k}_j) c_{ij} + \left(1 - \frac{\langle k \rangle}{N} Q(\vec{k}_i, \vec{k}_j)\right) (1 - c_{ij})} \right]^{-1} \\ &= \left[1 + \frac{n(\mathbf{c})}{n(\mathbf{c}')} \prod_{ij} \left(\frac{\frac{\langle k \rangle}{N} Q(\vec{k}_i, \vec{k}_j)}{1 - \frac{\langle k \rangle}{N} Q(\vec{k}_i, \vec{k}_j)} \right)^{c'_{ij} - c_{ij}} \right]^{-1} \end{aligned} \quad (29)$$

If the proposed move is a *square edge swap*, it is characterized by four distinct nodes (i, j, k, ℓ) , and takes us from a graph \mathbf{c}' with $c'_{ij} c'_{k\ell} \bar{c}'_{kj} \bar{c}'_{i\ell} = 1$ to a new graph \mathbf{c} with $\bar{c}_{ij} \bar{c}_{k\ell} c_{kj} c_{i\ell} = 1$ (leaving all other $N^2 - 4$ bond variables unaffected). For such moves the acceptance probabilities (29) become

$$A_\square(\mathbf{c}|\mathbf{c}') = \left[1 + \frac{n(\mathbf{c})}{n(\mathbf{c}')} \left(\frac{\frac{N}{\langle k \rangle Q(\vec{k}_i, \vec{k}_j)} - 1}{\frac{N}{\langle k \rangle Q(\vec{k}_k, \vec{k}_\ell)} - 1} \right) \left(\frac{\frac{N}{\langle k \rangle Q(\vec{k}_i, \vec{k}_j)} - 1}{\frac{N}{\langle k \rangle Q(\vec{k}_k, \vec{k}_\ell)} - 1} \right) \right]^{-1} \quad (30)$$

For large N we may choose to approximate this by

$$A_\square(\mathbf{c}|\mathbf{c}') \approx \left[1 + \frac{n(\mathbf{c})}{n(\mathbf{c}')} \frac{Q(\vec{k}_i, \vec{k}_j) Q(\vec{k}_k, \vec{k}_\ell)}{Q(\vec{k}_k, \vec{k}_j) Q(\vec{k}_i, \vec{k}_\ell)} \right]^{-1} \quad (31)$$

If the proposed move is a *triangle edge swap*, it is characterized by three distinct nodes (i, j, k) , and takes us from a

graph \mathbf{c}' with $c'_{ij}c'_{jk}c'_{kl}\bar{c}'_{ji}\bar{c}'_{kj}\bar{c}'_{ik} = 1$ to a new graph \mathbf{c} with $\bar{c}_{ij}\bar{c}_{jk}\bar{c}_{ki}c_{ji}c_{kj}c_{ik} = 1$ (leaving all other $N^2 - 6$ bond variables unaffected). Now the acceptance probabilities (29) become

$$A_{\Delta}(\mathbf{c}|\mathbf{c}') = \left[1 + \frac{n(\mathbf{c})}{n(\mathbf{c}')} \left(\frac{N}{\langle k \rangle Q(\vec{k}_i, \vec{k}_i)} - 1 \right) \left(\frac{N}{\langle k \rangle Q(\vec{k}_j, \vec{k}_j)} - 1 \right) \left(\frac{N}{\langle k \rangle Q(\vec{k}_k, \vec{k}_k)} - 1 \right) \right]^{-1} \quad (32)$$

For large N we may choose to approximate this by

$$A_{\Delta}(\mathbf{c}|\mathbf{c}') = \left[1 + \frac{n(\mathbf{c})}{n(\mathbf{c}')} \frac{Q(\vec{k}_i, \vec{k}_j)Q(\vec{k}_j, \vec{k}_k)Q(\vec{k}_k, \vec{k}_i)}{Q(\vec{k}_j, \vec{k}_i)Q(\vec{k}_k, \vec{k}_j)Q(\vec{k}_i, \vec{k}_k)} \right]^{-1} \quad (33)$$

VI. NUMERICAL SIMULATIONS OF THE CANONICAL RANDOMIZATION PROCESS

In this section we describe numerical simulations of our canonical MCMC graph randomization process and its ‘accept all’ edge swapping counterpart, applied to synthetic networks and to biological signalling networks. The most convenient marker of sampling bias in randomisation is the mobility $n(\mathbf{c})$ itself, which we will therefore use as to monitor the dynamics of the process. We used the Mersenne Twister random number generator from [21]. For numerical implementation, we use expressions for the incremental change in the mobility terms following a single edge swap move (similar to how this was done for nondirected networks [16]) – see appendix A. This avoids having to calculate $n(\mathbf{c})$ after each move, which would involve repeated matrix multiplications. Full source code (in C++) and Windows executables of our implementation are available on request.

A. Split flow network

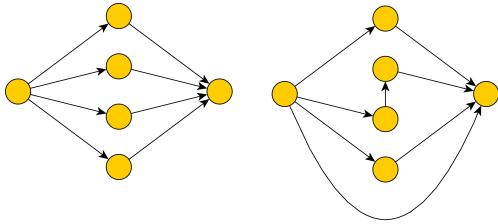


FIG. 3. The possible realisations of a split flow type network, with $N = K + 2$. The left hand configuration has a mobility of $K(K - 1)$; there is only one such configuration. The right hand configuration has mobility of $2K - 3$; there are $K(K - 1)$ such configurations.

A split flow network, see e.g. [17], is built as follows. Node $i = 1$ has degrees $(k_1^{\text{in}}, k_1^{\text{out}}) = (0, K)$, we have K nodes ($i = 2 \dots K + 1$) with degrees $(k_i^{\text{in}}, k_i^{\text{out}}) = (1, 1)$, and a final node with degrees $(k_{K+2}^{\text{in}}, k_{K+2}^{\text{out}}) = (K, 0)$. There exist two types of graph with this specified degree sequence. The first is shown

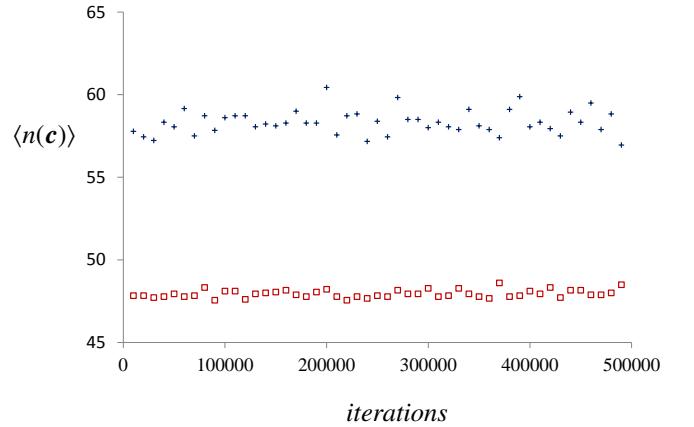


FIG. 4. Comparison for ‘split flow’ networks with $K = 25$ of randomization via ‘accept all’ edge swapping (squares) versus edge swapping with the canonical acceptance probabilities (crosses). The mobility $\langle n(\mathbf{c}) \rangle$ is used as a dynamical observable, since its expectation value is sensitive to sampling bias. Each marker gives the average mobility over 10,000 iterations. Observed values are in good agreement with theoretical predictions: $\langle n(\mathbf{c}) \rangle \approx 58.32$ for ‘accept all’ edge swapping (predicted: 58.52), versus $\langle n(\mathbf{c}) \rangle \approx 47.95$ for correct edge swapping (predicted: 47.92).

in the left of figure 3. The second type is obtained from the first by choosing two of the K ‘inner nodes’, of which one will cease to receive a link from $i = 1$ and the second will cease to provide a link to $i = K + 2$; so the mobility of the left graph is $n(\mathbf{c}) = K(K - 1)$. On the right-hand side configurations in figure 3 we can execute three possible square edge swap types: returning to the previous state (1 move), changing the internal node that is not receiving a link from $i = 1$ ($K - 2$ moves), or changing the internal node that is not sending a link to $i = K + 2$ ($K - 2$ moves), giving a total mobility for the graphs on the right of $n(\mathbf{c}) = 2K - 3$. The total number of such split flow networks is $|\Omega| = K(K - 1) + 1$.

Figure 4 shows graph randomisation dynamics for a split-flow network with $K = 25$, comparing ‘accept all’ edge swapping (which would sample graphs with the bias $p(\mathbf{c}) = n(\mathbf{c}) / \sum_{\mathbf{c}' \in \Omega} n(\mathbf{c}')$) to the canonical edge swap process (21) that is predicted to give unbiased sampling of graphs $p(\mathbf{c}) = 1/|\Omega|$. The predicted expectation values of the mobilities in the two sampling protocols are

$$\text{‘accept all’}: \langle n(\mathbf{c}) \rangle = \frac{\sum_{\mathbf{c} \in \Omega} n^2(\mathbf{c})}{\sum_{\mathbf{c} \in \Omega} n(\mathbf{c})} = \frac{5K^2 - 13K + 9}{2(K-1)} \approx 58.52$$

$$\text{canonical: } \langle n(\mathbf{c}) \rangle = \frac{\sum_{\mathbf{c} \in \Omega} n(\mathbf{c})}{|\Omega|} = \frac{2K(K-1)^2}{1+K(K-1)} \approx 47.92$$

The simulation results confirm these quantitative predictions (see caption of figure 4 for details), and underline the sampling bias caused by ‘accept all’ edge swapping, as well as the lack of such a bias in our canonical MCMC process.

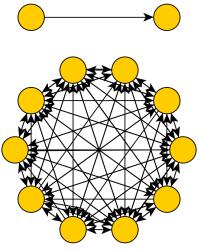


FIG. 5. The directed version of a ‘nearly hardcore’ network. Given the imposed degree sequences, there are only two types of graphs: the one shown here, and the one obtained by via an edge swap that involves the nodes of the isolated link and two nodes from the core.

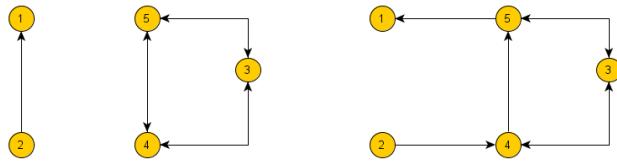


FIG. 6. Illustration of the edge swap that transforms a nearly hardcore graph from state A to one of the type B states.

B. Nearly hardcore networks

‘Nearly hardcore’ networks are another example of graphs for which ‘accept all’ edge swap sampling are known to exhibit a significant bias [16]. The directed version of such networks is constructed from a single isolated bond plus a complete subgraph of size $K = N - 2$. See figure 5. Triangle swaps are not possible. From the graph shown in the figure (the ‘mobile’ state, A) there are $K(K-1)$ ways to choose two nodes of the core to combine with the two non-core nodes to form an edge swap quartet, hence this state has $n_A(\mathbf{c}) = K(K-1)$. After an edge swap the graph in figure 5 is replaced by one in which one non-core node receives a link from the core, and the other sends a link to the core; see figure 6. There are $K(K-1)$ such graphs, to be called type B, hence the total number of nearly hardcore graphs is $|\Omega| = K(K-1)+1$. From each type B graph the inverse swap can be applied, plus $2(K-2)$ further moves that each equate to replacement of one of the core nodes involved in the previous swap by another. Hence $n_B(\mathbf{c}) = 2K-3$. These statements are confirmed by formula (9).

The predicted expectation values of the mobilities in the two sampling protocols, ‘accept all’ edge swapping (which would sample graphs with the bias $p(\mathbf{c}) = n(\mathbf{c}) / \sum_{\mathbf{c}' \in \Omega} n(\mathbf{c}')$) and the canonical edge swap process (21) (predicted to give unbiased sampling of graphs $p(\mathbf{c}) = 1/|\Omega|$), are

$$\text{‘accept all’: } \langle n(\mathbf{c}) \rangle = \frac{n_A^2(\mathbf{c}) + K(K-1)n_B^2(\mathbf{c})}{n_A(\mathbf{c}) + K(K-1)n_B(\mathbf{c})} = \frac{5K^2 - 13K + 9}{2(K-1)}$$

$$\text{canonical: } \langle n(\mathbf{c}) \rangle = \frac{n_A(\mathbf{c}) + K(K-1)n_B(\mathbf{c})}{1 + K(K-1)} = \frac{2K(K-1)^2}{1 + K(K-1)}$$

Figure 7 shows graph randomisation dynamics for a nearly hardcore network with $K = 18$ (so $N = 20$). Here the theory,

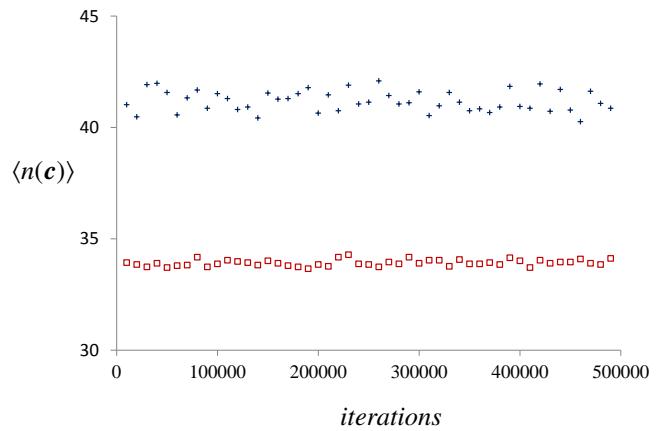


FIG. 7. Comparison for ‘nearly hardcore’ networks with $K = 18$ of randomization via ‘accept all’ edge swapping (squares) versus edge swapping with the canonical acceptance probabilities (crosses). Each marker gives the average mobility over 10,000 iterations. Observed mobility values are again in good agreement with theoretical predictions: $\langle n(\mathbf{c}) \rangle \approx 41.09$ for ‘accept all’ (predicted: 41.03), versus $\langle n(\mathbf{c}) \rangle \approx 33.92$ for correct edge swapping (predicted: 33.89).

i.e. the previous two formulae, predicts that we should see $\langle n(\mathbf{c}) \rangle \approx 41.03$ for ‘accept all’ edge swapping, and $\langle n(\mathbf{c}) \rangle \approx 33.89$ for unbiased sampling. Again the simulation results confirm our predictions (see caption of figure 7 for details).

C. Application to gene regulation networks

Gene regulation networks are important examples of directed biological networks. Figures 8 and 9 show numerical results of the randomization dynamics applied to the gene regulation network data of [22] (with $N = 5654$ nodes) and [23] (with $N = 3865$ nodes), respectively. We apply all three randomization processes discussed so far in this paper, viz. ‘accept all’ edge swapping, canonical edge swapping aimed at uniform sampling of all graphs with the degree sequences of the biological network, and canonical edge swapping aimed at uniform sampling of all graphs with the degree sequence $(\vec{k}_1, \dots, \vec{k}_N)$ and (on average) the degree-degree correlation kernel $W(\vec{k}, \vec{k}')$ of the biological network. In contrast to the synthetic examples in the previous subsection, in gene regulation networks we do not observe significant divergence between ‘accept all’ versus canonical edge swap randomization; this is similar to what was observed earlier for the randomization of protein-protein interaction networks in [16]. We also see that in both cases the biological network is significantly more mobile than the typical network with the same degree sequence. However, figure 8 suggests that the set of networks that share with the biological one both the degree sequence and the degree correlations (and hence resemble more closely the biological network under study) all have high mobilities.

Implementing degree-degree correlation targeting directly has the effect of severely reducing the space of graphs through which the process can pass, hence we would expect

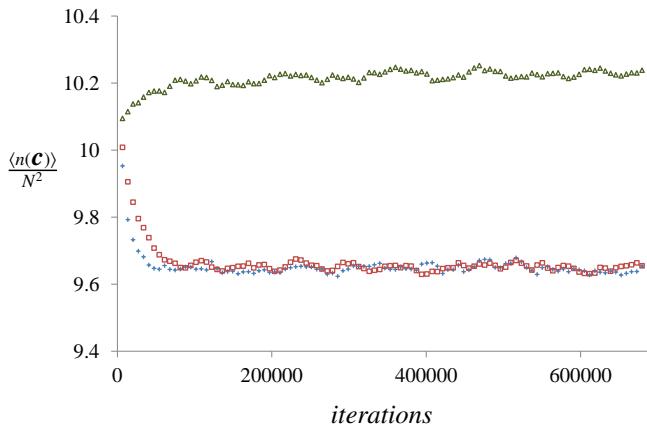


FIG. 8. Randomization dynamics for the gene regulation network data of [22]. The observable shown is a rolling average of the normalized average square mobility $\langle n_{\square}(c) \rangle / N^2$. We compare ‘accept all’ edge swapping (+), canonical edge swapping aimed at uniform sampling of all graphs with the biological degree sequence of the biological network (\square), and canonical edge swapping aimed at uniform sampling of all graphs with the degree sequence $(\vec{k}_1, \dots, \vec{k}_N)$ and the degree-degree correlation kernel $W(\vec{k}, \vec{k}')$ of the biological network (\triangle). Hamming distances between the start and end networks of \square , + and \triangle were 0.8, 0.8 and 0.75 respectively.

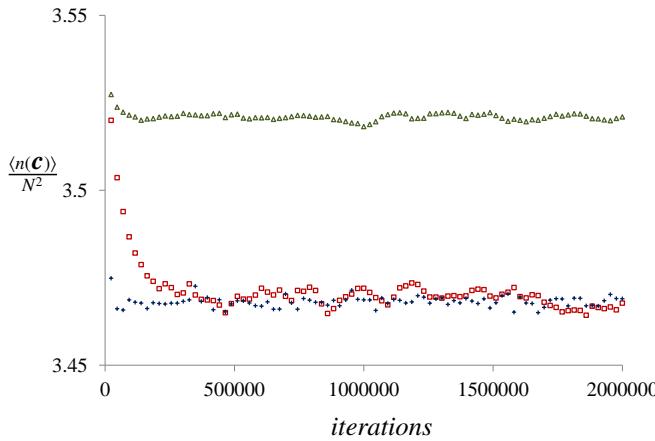


FIG. 9. Randomization dynamics for the gene regulation network of [23]. The key and the axes are the same as in figure 8. Hamming distances between the start and end networks of \square , + and \triangle were 0.94, 0.94 and 0.86 respectively.

finite-size effects to be more pronounced. The process would be less restricted, and hence more natural, with a smoothed target degree-degree correlation. There is a trade-off between the flexibility of the process and the accuracy of the targeting. We have used a light Gaussian smoothing, generalising what was used in [24] to the higher dimension we need. The best choice target degree-degree correlations - including decisions about smoothing - will very much depend on the particular problem being studied.

D. Targeting degree-degree correlation

In addition to being unbiased, the canonical MCMC process in this paper can sample according to any specified measure on the space of degree-constrained graphs. The particular example which we’ve developed is the generation of directed graphs from the tailored ensemble (28), via the acceptance probabilities (30,32). Figures 8 and 9 show the trajectory of this process for two different datasets. Figure 10 is provided to illustrate that the network corresponding to this process successfully reproduces the key features of the assortativity of the real network. In particular, the characteristic downwards slope was postulated by [10] to be a key feature of protein networks, associated with greater stability and improved specificity.

VII. CONCLUSION

In this paper we have built on the work of [3] and [16] to define an ergodic and unbiased stochastic process for randomising directed binary non-self-interacting networks, which keeps the number of in- and out- connections of each node constant. The result takes the form of a canonical Markov Chain Monte Carlo (MCMC) algorithm based on simple direct edge swaps and triangle reversals, with nontrivial move acceptance probabilities that are calculated from the current state of the network only. The acceptance probabilities correct for the entropic bias in ‘accept all’ edge-swap randomization, which is caused by the state dependence of the number of moves that can be executed (the ‘mobility’ of a graph).

Our process is precise for any network size and network topology, and sufficiently versatile to allow random directed graphs with the correct in- and out-degree sequence to be generated with arbitrary desired sampling probabilities. The algorithm can be used e.g. to generate truly unbiased random directed graphs with imposed degrees for hypothesis testing (in contrast to the ‘edge stub’ algorithm or the ‘accept all’ edge swap algorithm, both of which are biased), or to generate more sophisticated null models which inherit from a real network both the degree sequence *and* the degree correlations, but are otherwise random and unbiased.

Our core insight is similar to [25] and [3]. However, our work takes the formalism further, and generates a direct adjustment to the MCMC based on the current state of the network only, rather than a retrospective adjustment to the observed process [25] or a search of the entire state-space [3]. Moreover, our approach can be generalised to generate more tailored null-models (e.g. our example of targeting a specified degree-degree correlation).

We have derived bounds to predict for which degree sequences the differences between ‘accept all’ and correct randomization (i.e. the effects of sampling bias) are negligible. Application to synthetic networks showed a large discrepancy between the ‘accept all’ and correct randomization processes, and good agreement with our theoretical predictions for the values of key observables that are affected by the entropic bias of incorrect randomization. For the biological networks which we studied (gene regulation networks) we find the differences

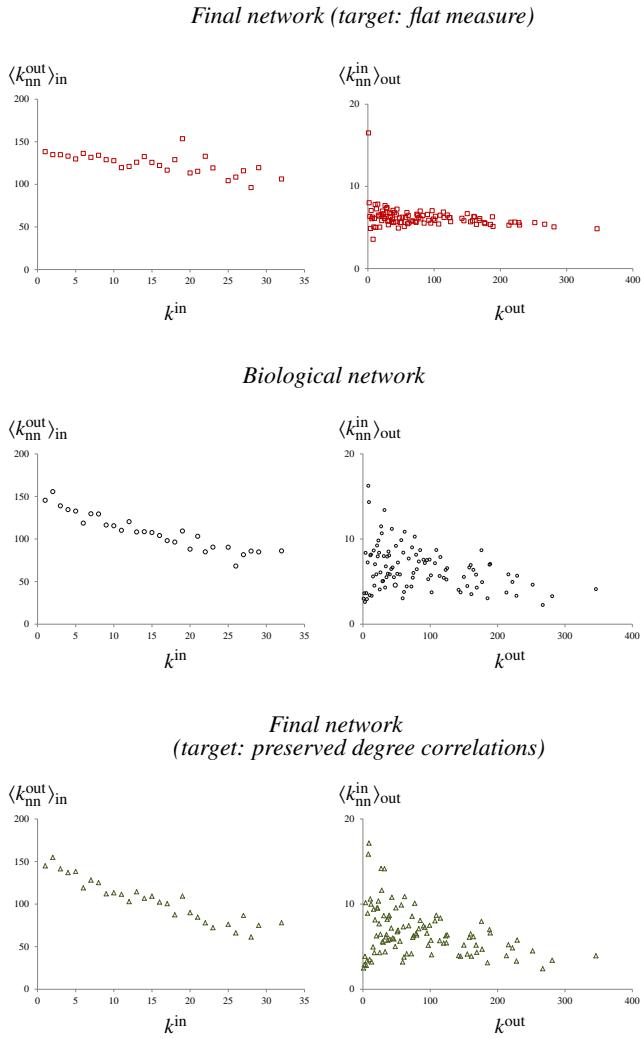


FIG. 10. These charts summarize the degree-degree correlations observed in the original network (middle row \circ), the final network after the process targeting the flat measure (top row \square) and the process tailored to preserve the degree-degree correlation pattern of the original network (bottom row Δ). The data used is based on [23] and the process shown in figure 9. The left hand charts summarize the correlation between the in-degree of a node and the average out-degree $\langle k_{nn}^{out} \rangle_{in}$ of its in-neighbours. The right hand charts summarize the correlation between the out-degree of a node and the average in-degree $\langle k_{nn}^{in} \rangle_{out}$ of its out-neighbours. This representation was chosen as a widely adopted and easy to interpret measure of the assortativity of a directed network.

between correct and incorrect sampling in the space of graphs with imposed degree sequences to be negligible. However, this cannot be relied upon to continue in future studies, especially when network datasets become less sparse, or randomization processes which target more complicated topological observables are used.

Biological signalling networks tend to have ‘fat-tailed’ distributions with low average degree and relatively high clustering levels, whereas in a graph ensemble defined by prescrib-

ing in- and out- degree sequences and uniform graph probabilities, graphs will typically have $O(1)$ triangles per node or less. Hence, if we run edge swap processes on such ensembles, by the time equilibration is approached the algorithm will typically be moving through networks with low clustering, where the change in mobility coming from those terms that ‘count’ triangles will be very low. However, this will be different if we target a non-flat measure, for instance if we generate graphs with degree-degree correlations. Since biological degree-degree correlations seem to be associated with clustering, it will become increasingly dangerous to assume that the sampling bias caused by using ‘accept all’ edge swap dynamics will be modest.

Given that precise and practical alternatives are now available, we feel that there is no justification for the use of biased graph randomization processes. In those cases where we seek to generate unbiased random directed graphs with in- and out-degrees identical to some observed network, our canonical MCMC process would take the observed graph as its seed and take care of the required unbiased sampling. In those cases where we specify degree sequences *ab initio*, without having a seed graph, one may use the Molloy-Reed algorithm to generate a (biased) seed graph prior to running our algorithm.

In addition to being rigorously free of entropic sampling bias, our present canonical MCMC process is also able to generate directed degree-constrained networks with any arbitrary specified sampling probabilities. We have shown examples of the generation of synthetic graphs generated with precisely controlled expectation values for the degree-degree correlation kernels, where the imposed sampling measure is a maximum entropy distribution on the set of graphs with prescribed degrees, with degree correlations imposed as a soft constraint. Degree correlation is a promising candidate to define a better null model, as it has been observed in the literature to act as a ‘signature’ distinguishing different types of networks (e.g. [26, 27]).

Two directions for future research could be to look at weighted networks (e.g. to integrate our ideas with those in papers such as [28]), or at bipartite networks (which also have interesting applications, see e.g. [29]). Furthermore, it would seem appropriate in the field of network hypothesis testing to take more seriously the nontrivial number of short loops in biological signalling systems. Whenever we randomize within the large amorphous space of graphs that inherit from the biological network only the degree sequence, we are effectively running a dynamics on graphs that are locally tree-like, where (conveniently) the mobility issues are minor. But we know already that this large set will typically produce null models that are very much unlike biological networks, for that same reason. How informative are small p-values in this context?

ACKNOWLEDGEMENTS

This study was supported by the Biotechnology and Biological Sciences Research Council of the United Kingdom. It is our pleasure to thank Thomas Schlitt for providing gene regulation network data.

[1] M. Molloy and B. Reed, Random Structures & Algorithms **6**, 161 (1995), <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.6195>

[2] R. Albert and A. L. Barabási, Reviews of Modern Physics **74**, 47 (Jan. 2002), <http://dx.doi.org/10.1103/RevModPhys.74.47>

[3] A. R. Rao, R. Jana, and S. Bandyopadhyay, Sankhy: The Indian Journal of Statistics, Series A **58** (1996), ISSN 0581572X, doi: “bibinfo doi 10.2307/25051102, <http://dx.doi.org/10.2307/25051102>

[4] T. Squartini, G. Fagiolo, and D. Garlaschelli, Phys. Rev. E **84**, 046118 (Oct 2011), <http://link.aps.org/doi/10.1103/PhysRevE.84.046118>

[5] T. Squartini, G. Fagiolo, and D. Garlaschelli(2011), <http://arxiv.org/abs/1103.1243>

[6] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, Nature Genetics **31**, 64 (Apr. 2002), ISSN 1061-4036, <http://dx.doi.org/10.1038/ng881>

[7] G. A. Pagani and M. Aiello, “The power grid as a complex network: a survey,” (May 2011), <http://arxiv.org/abs/1105.3338>

[8] T. Ohnishi, H. Takayasu, and M. Takayasu, Journal of Economic Interaction and Coordination(Jun. 2010), ISSN 1860-711X, doi:“bibinfo doi 10.1007/s11403-010-0066-6, <http://dx.doi.org/10.1007/s11403-010-0066-6>

[9] M. E. J. Newman, D. J. Watts, and S. H. Strogatz, Proceedings of the National Academy of Sciences of the United States of America **99**, 2566 (Feb. 2002), ISSN 0027-8424, <http://dx.doi.org/10.1073/pnas.012582999>

[10] S. Maslov and K. Sneppen, Science **296**, 910 (May 2002), ISSN 1095-9203, <http://dx.doi.org/10.1126/science.1065103>

[11] N. J. Gotelli and W. Ulrich, Oikos, no(2011), ISSN 1600-0706, <http://dx.doi.org/10.1111/j.1600-0706.2011.20301.x>

[12] O. D. King, Phys. Rev. E **70**, 058101+ (2004)

[13] H. Klein-Hennig and A. K. Hartmann, “Bias in generation of random graphs,” (2011), <http://arxiv.org/abs/1107.5734>

[14] H. Kim, C. I. Del Genio, K. E. Bassler, and Z. Toroczkai(Sep. 2011), arXiv:1109.4590, <http://arxiv.org/abs/1109.4590>

[15] S. Itzkovitz, R. Milo, N. Kashtan, G. Ziv, and U. Alon, Physical Review E **68**, 026127+ (Aug. 2003), <http://dx.doi.org/10.1103/PhysRevE.68.026127>

[16] A. C. C. Coolen, A. De Martino, and A. Annibale, J. Stat. Phys.(May 2009), arXiv:0905.4155

[17] R. Milo, N. Kashtan, S. Itzkovitz, M. E. J. Newman, and U. Alon, “On the uniform generation of random graphs with prescribed degree sequences,” (May 2004), arXiv:cond-mat/0312028, <http://arxiv.org/abs/cond-mat/0312028>

[18] J. J. Seidel, in *In Colloquio Internazionale sulle Teorie Combinatorie Tomo I*, edited by A. Doe (1973) pp. 481-511

[19] R. Taylor, *Combinatorial Mathematics VIII* (Springer, 1981)

[20] E. S. Roberts, T. Schlitt, and A. C. C. Coolen, J. Phys. A **44**, 275002 (2011), <http://stacks.iop.org/1751-8121/44/i=27/a=275002>

[21] M. Matsumoto and T. Nishimura, ACM Trans. Model. Comput. Simul. **8**, 3 (Jan. 1998), ISSN 1049-3301, <http://dx.doi.org/10.1145/272991.272995>

[22] T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburty, J. Simon, M. Bard, and S. H. Friend, Cell **102**, 109 (Jul. 2000), ISSN 0092-8674, <http://view.ncbi.nlm.nih.gov/pubmed/10929718>

[23] C. T. Harbison, D. B. Gordon, T. I. I. Lee, N. J. Rinaldi, K. D. Macisaac, T. W. Danford, N. M. Hannett, J.-B. B. Tagne, D. B. Reynolds, J. Yoo, E. G. Jennings, J. Zeitlinger, D. K. Pokholok, M. Kellis, P. A. Rolfe, K. T. Takusagawa, E. S. Lander, D. K. Gifford, E. Fraenkel, and R. A. Young, Nature **431**, 99 (Sep. 2004), ISSN 1476-4687, <http://dx.doi.org/10.1038/nature02800>

[24] L. P. Fernandes, A. Annibale, J. Kleinjung, A. C. C. Coolen, and F. Fraternali, PLoS ONE **5**, e12083+ (Aug. 2010), ISSN 1932-6203, <http://dx.doi.org/10.1371/journal.pone.0012083>

[25] Y. Artzy-Randrup and L. Stone, Phys. Rev. E **72**, 056708 (Nov 2005), <http://link.aps.org/doi/10.1103/PhysRevE.72.056708>

[26] S. Maslov, K. Sneppen, and A. Zaliznyak, Physica A: Statistical Mechanics and its Applications **333**, 529 (2004), ISSN 0378-4371, <http://www.sciencedirect.com/science/article/pii/S0378437103008409>

[27] M. E. J. Newman, Phys. Rev. Lett. **89**, 208701 (Oct 2002), <http://link.aps.org/doi/10.1103/PhysRevLett.89.208701>

[28] G. Ansmann and K. Lehnertz, Phys. Rev. E **84**, 026103+ (Aug. 2011), <http://dx.doi.org/10.1103/PhysRevE.84.026103>

[29] G. Basler, O. Ebenhöh, J. Selbig, and Z. Nikoloski, Bioinformatics(Mar. 2011), ISSN 1367-4811, doi:“bibinfo doi 10.1093/bioinformatics/btr145, <http://dx.doi.org/10.1093/bioinformatics/btr145>

Appendix A: Efficient calculation of changes in mobility terms following one move

Calculating the mobility $n(c)$ terms is computationally heavy. Given that our moves are simple and standard, we follow the alternative route in [16] and derive formulae for calculating the *change* in mobility due to one move, so that we can avoid repeated heavy matrix multiplications at each time step.

1. Change in $n_{\square}(c)$ following one square-type move

Without loss of generality, define our square move to be the transformation between matrix c and x , involving four nodes (a, b, c, d) , such that for all (i, j) : $x_{ij} = c_{ij} + \Delta_{ij}$, with

$$\Delta_{ij} = \delta_{ia}\delta_{jd} + \delta_{ic}\delta_{jb} - \delta_{ia}\delta_{jb} - \delta_{ic}\delta_{jd}$$

We now determine the overall change induced in $n_{\square}(c)$ by finding the impact of an edge swap on each term in (9). on the right hand side of the expression above.

- Term 1:

$$\begin{aligned} \text{Tr}(\mathbf{x}\mathbf{x}^\dagger \mathbf{x}\mathbf{x}^\dagger) - \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}\mathbf{c}^\dagger) &= \sum_{ijkm} \left[c_{ij}c_{kj}c_{km}c_{im} \right. \\ &\quad \left. - (c_{ij} + \Delta_{ij})(c_{kj} + \Delta_{kj})(c_{km} + \Delta_{km})(c_{im} + \Delta_{im}) \right] \\ &= \Delta_{ij}c_{kj}c_{km}c_{im} + \dots + \Delta_{ij}\Delta_{kj}c_{km}c_{im} + \dots \\ &\quad + \Delta_{ij}\Delta_{kj}\Delta_{km}c_{im} + \dots + \Delta_{ij}\Delta_{kj}\Delta_{km}\Delta_{im} \end{aligned}$$

where ... refers in each case to three similar terms (with their appropriate indices). Let us inspect what happens when two Δ terms are multiplied together. We might have the first suffix repeated, the second suffix repeated, or no repeated sufficies:

$$\begin{aligned} \Delta_{ij}\Delta_{im} &= 2[\delta_{jd}(\delta_{md} - \delta_{mb}) + \delta_{jb}(\delta_{mb} - \delta_{md})] \\ \Delta_{ij}\Delta_{kj} &= 2[\delta_{ia}(\delta_{ka} - \delta_{kc}) + \delta_{ic}(\delta_{kc} - \delta_{ka})] \end{aligned} \quad (\text{A1})$$

One immediately observes that

$$\begin{aligned} \sum_{ijkm} \Delta_{ij}\Delta_{kj}\Delta_{km}\Delta_{im} &= 4 \sum_{ik} [\delta_{ia}\delta_{ia}(\delta_{ka}\delta_{ka} + \delta_{kc}\delta_{kc}) + \delta_{ic}\delta_{ic}(\delta_{kc}\delta_{kc} + \delta_{ka}\delta_{ka})] \\ &= 16 \end{aligned}$$

To handle two Δ terms with different sufficies we use

$$\Delta_{ij}c_{kj} = c_{kb}(\delta_{ic} - \delta_{ia}) + c_{kd}(\delta_{ia} - \delta_{ic}) \quad (\text{A2})$$

which leads us to

$$\sum_{ijkm} \Delta_{ij}c_{kj}\Delta_{km}c_{im} = 4$$

Returning to the result A1 it follows that

$$\begin{aligned} \Delta_{ij}\Delta_{kj}c_{im}c_{km} &= 2(\delta_{ia}(\delta_{ka} - \delta_{kc}) + \delta_{ic}(\delta_{kc} - \delta_{ka}))c_{im}c_{km} \\ &= 2(k_a^{\text{out}} + k_c^{\text{out}}) - 4c_{am}c_{cm} \end{aligned}$$

and the symmetric term gives

$$\Delta_{ij}\Delta_{im}c_{kj}c_{km} = 2(k_d^{\text{in}} + k_b^{\text{in}}) - 4c_{id}c_{ib}$$

For the third order terms we combine (A1) and (A2):

$$\begin{aligned} \sum_{ijkm} \Delta_{ij}\Delta_{im}\Delta_{kj}c_{km} &= 2 \sum_{ik} \left[[\delta_{ia}(\delta_{ka} - \delta_{kc}) + \delta_{ic}(\delta_{kc} - \delta_{ka})] \right. \\ &\quad \left. \times [\delta_{ia}c_{kd} + \delta_{ic}c_{kb} - \delta_{ia}c_{kb} - \delta_{ic}c_{kd}] \right] \\ &= 2(c_{ad} - c_{cd} - c_{ab} + c_{cb} - c_{ab} + c_{cb} + c_{ad} - c_{cd}) \\ &= -8 \end{aligned}$$

By permutation of sufficies all such terms evaluate to -8 . Finally we turn to the four terms where only one Δ appears, corresponding to permutations of

$\Delta_{ij}c_{kj}c_{km}c_{im} = c_{kd}c_{km}c_{am} + c_{kb}c_{km}c_{cm} - c_{kb}c_{km}c_{am} - c_{kd}c_{km}c_{cm}$. Adding up all separate elements above, we obtain the change in the square mobility term due to one application of a square move:

$$\begin{aligned} \Delta \left[\frac{1}{2} \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}\mathbf{c}^\dagger) \right] &= 2(k_b^{\text{in}} + k_b^{\text{out}} + k_a^{\text{out}} + k_c^{\text{out}}) \\ &\quad + 2(c_{kd}c_{km}c_{am} + c_{kb}c_{km}c_{cm} - c_{kb}c_{km}c_{am} - c_{kd}c_{km}c_{cm}) \\ &\quad - 4(c_{id}c_{ib} + c_{am}c_{cm} + 1) \end{aligned} \quad (\text{A3})$$

- Term 2:

$$\begin{aligned} \Delta \left[\sum_{ij} k_i^{\text{out}} c_{ij} k_j^{\text{in}} \right] &= \sum_{ij} k_i^{\text{out}} (x_{ij} - c_{ij}) k_j^{\text{in}} \\ &= \sum_{ij} k_i^{\text{out}} [\delta_{ia}\delta_{jd} + \delta_{ic}\delta_{jb} - \delta_{ia}\delta_{jb} - \delta_{ic}\delta_{jd}] k_j^{\text{in}} \\ &= k_a^{\text{out}} k_d^{\text{in}} + k_c^{\text{out}} k_b^{\text{in}} - k_a^{\text{out}} k_b^{\text{in}} - k_c^{\text{out}} k_d^{\text{in}} \end{aligned} \quad (\text{A4})$$

- Term 3:

$$\begin{aligned} \text{Tr}(\mathbf{x}\mathbf{x}^\dagger \mathbf{x}) - \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}) &= \sum_{ijk} [c_{ij} + \Delta_{ij}][c_{kj} + \Delta_{kj}][c_{ki} + \Delta_{ki}] - c_{ij}c_{kj}c_{ki} \\ &= \sum_{ijk} [\Delta_{kj}c_{ij}c_{ki} + \Delta_{ij}c_{kj}c_{ki} + \Delta_{ki}c_{ij}c_{kj} + \Delta_{ij}\Delta_{kj}c_{ki} \\ &\quad + \Delta_{kj}\Delta_{ki}c_{ij} + \Delta_{ij}\Delta_{ki}c_{kj} + \Delta_{ij}\Delta_{kj}\Delta_{ki}] \end{aligned}$$

The product of two Δ terms gives

$$\begin{aligned} \Delta_{ij}\Delta_{kj} &= \delta_{jd}(\delta_{ia} - \delta_{ic}) + \delta_{jb}(\delta_{ic} - \delta_{ia}) \\ &\quad - \delta_{jd}(\delta_{ia}\delta_{kc} + \delta_{ic}\delta_{ka}) - \delta_{jb}(\delta_{ic}\delta_{ka} + \delta_{ia}\delta_{kc}) \end{aligned}$$

but $\Delta_{ij}\Delta_{ki} = 0$, and in a straightforward way we obtain

$$\begin{aligned} \sum_{ijk} \Delta_{kj}c_{ij}c_{ki} &= \sum_{ijk} [\delta_{ka}\delta_{jd} + \delta_{kc}\delta_{jb} - \delta_{ka}\delta_{jb} - \delta_{kc}\delta_{jd}] c_{ij}c_{ki} \\ &= \sum_i [c_{ai}c_{id} + c_{ci}c_{ib} - c_{ai}c_{ib} - c_{ci}c_{id}] \end{aligned}$$

Assembling all terms and their symmetric equivalents leads to an expression which can be summarised as

$$\begin{aligned} \Delta \left[\text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}) \right] &= \text{MutN}(a, d) + \text{MutN}(c, b) - \text{MutN}(a, b) \\ &\quad - \text{MutN}(c, d) - 2(c_{bd} + c_{db} + c_{ac} + c_{ca}) \end{aligned} \quad (\text{A5})$$

where

$$\text{MutN}(\alpha, \beta) = \sum_i [c_{\alpha i}c_{i\beta} + c_{\alpha i}c_{\beta i} + c_{i\alpha}c_{i\beta}] \quad (\text{A6})$$

- Term 5:

$$\begin{aligned}
\Delta [\text{Tr}(\mathbf{c}^2)] &= \text{Tr}(\mathbf{x}^2) - \text{Tr}(\mathbf{c}^2) \\
&= \sum_{ij} [c_{ij} + (\delta_{ia}\delta_{jd} + \delta_{ic}\delta_{jb} - \delta_{ia}\delta_{jb} - \delta_{ic}\delta_{jd})] \\
&\quad \times [c_{ji} + (\delta_{ja}\delta_{id} + \delta_{jc}\delta_{ib} - \delta_{ja}\delta_{ib} - \delta_{jc}\delta_{id})] - \text{Tr}(\mathbf{c}^2) \\
&= 2(c_{da} + c_{bc} - c_{ba} - c_{dc}) \tag{A7}
\end{aligned}$$

- Terms 4 and 6:

The two terms $\frac{1}{2}N^2\langle k \rangle^2$ and $\sum_i k_i^{\text{out}} k_i^{\text{in}}$ do not change, since our stochastic process conserves all degrees.

In combination, the above ingredients lead us to the following update formula for the square mobility (9), as a result of the edge swap (A1):

$$\begin{aligned}
\Delta n_{\square} &= 2(k_d^{\text{in}} + k_b^{\text{in}} + k_a^{\text{out}} + k_c^{\text{out}}) \\
&\quad + 2(c_{kd}c_{km}c_{am} + c_{kb}c_{km}c_{cm} - c_{kb}c_{km}c_{am} - c_{kd}c_{km}c_{cm}) \\
&\quad - 4(c_{id}c_{ib} + c_{am}c_{cm} + 1) \\
&\quad - [k_a^{\text{out}}k_d^{\text{in}} + k_c^{\text{out}}k_b^{\text{in}} - k_a^{\text{out}}k_b^{\text{in}} - k_c^{\text{out}}k_d^{\text{in}}] \\
&\quad + \text{MutN}(a, d) + \text{MutN}(c, b) - \text{MutN}(a, b) \\
&\quad - \text{MutN}(c, d) - 2(c_{bd} + c_{db} + c_{ac} + c_{ca}) \\
&\quad + c_{da} + c_{bc} - c_{ba} - c_{dc} \tag{A8}
\end{aligned}$$

2. Change in $n_{\Delta}(\mathbf{c})$ following one square-type move

The different terms in the triangle mobility term (to be called Term 7, Term 8, Term 9 and Term 10, to avoid confusion with the previous section) are

$$n(\mathbf{c})_{\Delta} = \frac{1}{3}\text{Tr}(\mathbf{c}^3) - \text{Tr}(\mathbf{c}^{\dagger}\mathbf{c}^2) + \text{Tr}(\mathbf{c}^{\dagger 2}\mathbf{c}) - \frac{1}{3}\text{Tr}(\mathbf{c}^{\dagger 3})$$

- Term 7:

$$\begin{aligned}
\Delta \text{Tr}(\mathbf{c}^3) &= \sum_{ijk} [x_{ij}x_{jk}x_{ki} - c_{ij}c_{jk}c_{ki}] \\
&= 3 \sum_i (c_{ia}c_{di} + c_{ic}c_{bi} - c_{ia}c_{bi} - c_{ic}c_{di})
\end{aligned}$$

- Term 8:

Here we have to inspect first how the matrix \mathbf{c}^{\dagger} of double bonds is affected by a square move:

$$x_{ij}^{\dagger} = c_{ij}^{\dagger} + \Delta_{ij}^{\dagger} + \Delta_{ji}^{\dagger}$$

with

$$\Delta_{ij}^{\dagger} = \delta_{ia}\delta_{jd}c_{da} + \delta_{ic}\delta_{jb}c_{bc} - \delta_{ia}\delta_{jb}c_{ba} - \delta_{ic}\delta_{jd}c_{dc}$$

It follows that

$$\begin{aligned}
\Delta \text{Tr}(\mathbf{c}^{\dagger}\mathbf{c}^2) &= \text{Tr}(\mathbf{x}^{\dagger}\mathbf{x}^2) - \text{Tr}(\mathbf{c}^{\dagger}\mathbf{c}^2) \\
&= \sum_{ijk} (c_{ij}^{\dagger} + \Delta_{ij}^{\dagger} + \Delta_{ji}^{\dagger})(c_{jk} + \Delta_{jk}) (c_{ki} + \Delta_{ki}) - \text{Tr}(\mathbf{c}^{\dagger}\mathbf{c}^2)
\end{aligned}$$

Arguments similar to those employed before show that $\sum_j \Delta_{ij}^{\dagger} \Delta_{jk} = \sum_i \Delta_{ij}^{\dagger} \Delta_{ki} = 0$, whereas the remaining two ‘compound’ terms give

$$\begin{aligned}
\sum_{ijk} \Delta_{ji}^{\dagger} \Delta_{jk} c_{ki} &= \\
&\sum_{ijk} (\delta_{ja}\delta_{id}c_{da} + \delta_{jc}\delta_{ib}c_{bc} - \delta_{ja}\delta_{ib}c_{ba} - \delta_{jc}\delta_{id}c_{dc}) \\
&\quad \times (\delta_{ja}\delta_{kd} + \delta_{jc}\delta_{kb} - \delta_{ja}\delta_{kb} - \delta_{jc}\delta_{kd}) c_{ki} \\
&= (c_{da} + c_{dc})\delta_{id}(\delta_{kd} - \delta_{kb})c_{ki} \\
&\quad + (c_{bc} + c_{ba})\delta_{ib}(\delta_{kb} - \delta_{kd})c_{ki} \\
&= -(c_{da} + c_{dc})c_{bd} - (c_{bc} + c_{ba})c_{db}
\end{aligned}$$

and

$$\begin{aligned}
\sum_{ijk} \Delta_{ji}^{\dagger} c_{jk} \Delta_{ki} &= \\
&\sum_{ijk} (\delta_{ja}\delta_{id}c_{da} + \delta_{jc}\delta_{ib}c_{bc} - \delta_{ja}\delta_{ib}c_{ba} - \delta_{jc}\delta_{id}c_{dc}) \\
&\quad \times (\delta_{ka}\delta_{id} + \delta_{kc}\delta_{ib} - \delta_{ka}\delta_{ib} - \delta_{kc}\delta_{id}) c_{jk} \\
&= (c_{da} + c_{ba})\delta_{ja}(\delta_{ka} - \delta_{kc})c_{jk} \\
&\quad + (c_{bc} + c_{dc})\delta_{jc}(\delta_{kc} - \delta_{ka})c_{jk} \\
&= -[(c_{da} + c_{ba})c_{ac} + (c_{bc} + c_{dc})c_{ca}]
\end{aligned}$$

The product of three Deltas can be immediately seen to be zero by earlier arguments (repeated suffix in different positions). The other terms evaluate as follows:

$$\begin{aligned}
\sum_{ijk} \Delta_{ij}^{\dagger} c_{jk} c_{ki} &= \sum_{\beta \in \{a,c\}, \alpha \in \{d,b\}} \mathbb{I}(\alpha, \beta) c_{ak} c_{\alpha\beta} c_{k\beta} \tag{A9} \\
\sum_{ijk} \Delta_{ki} c_{ij}^{\dagger} c_{jk} &= \sum_{\beta \in \{a,c\}, \alpha \in \{d,b\}} \mathbb{I}(\alpha, \beta) c_{\alpha k}^{\dagger} c_{k\beta} \\
\sum_{ijk} \Delta_{jk} c_{ij}^{\dagger} c_{ki} &= \sum_{\beta \in \{a,c\}, \alpha \in \{d,b\}} \mathbb{I}(\alpha, \beta) c_{ak} c_{k\beta}^{\dagger}
\end{aligned}$$

where $\mathbb{I}(\alpha, \beta)$ is an indicator function which evaluates to 1 if bond (α, β) is created by the present move, to -1 if the bond (α, β) is destroyed, and zero otherwise. Similarly

$$\begin{aligned}
\sum_{ijk} \Delta_{ji}^{\dagger} c_{jk} c_{ki} &= \sum_{ijk} c_{jk} c_{ki} \\
&\quad \times (\delta_{ja}\delta_{id}c_{da} + \delta_{jc}\delta_{ib}c_{bc} - \delta_{ja}\delta_{ib}c_{ba} - \delta_{jc}\delta_{id}c_{dc}) \\
&= \sum_k (c_{ak}c_{kd}c_{da} + c_{ck}c_{kb}c_{bc} - c_{ak}c_{kb}c_{ba} - c_{ck}c_{kd}c_{dc})
\end{aligned}$$

Putting all of these sub-terms together yields:

$$\begin{aligned} \Delta[\text{Tr}(\mathbf{c}^\dagger \mathbf{c}^2)] &= \sum_{\beta \in \{a,c\}, \alpha \in \{d,b\}} \mathbb{I}(\alpha, \beta) \\ &\times \sum_k \left[c_{\alpha\beta} (c_{\alpha k} c_{k\beta} + c_{\beta k} c_{k\alpha}) + c_{\alpha k}^\dagger c_{k\beta} + c_{\alpha k} c_{k\beta}^\dagger \right] \\ &- c_{bd} (c_{da} + c_{dc}) + c_{ac} (c_{da} + c_{ba}) \\ &+ c_{db} (c_{bc} + c_{ba}) + c_{ca} (c_{bc} + c_{dc}) \end{aligned} \quad (\text{A10})$$

- Terms 9 and 10:

The same steps as followed to calculate term 8 can be also be applied to terms 9 and 10. In combination, the above ingredients lead us to the following update formula for the triangle mobility (10), as a result of the edge swap (A1):

$$\begin{aligned} \Delta n_\Delta &= \sum_{\beta \in \{1,3\}, \alpha \in \{4,2\}} \mathbb{I}(\alpha, \beta) \sum_k \left[c_{\alpha k} c_{k\beta} - c_{\alpha\beta} (c_{\alpha k} c_{k\beta} + c_{\beta k} c_{k\alpha}) \right. \\ &- c_{\alpha k}^\dagger c_{k\beta} - c_{\alpha k} c_{k\beta}^\dagger + c_{\alpha k}^\dagger c_{k\beta}^\dagger - c_{\alpha\beta} \left(c_{\alpha k}^\dagger c_{k\beta}^\dagger + c_{k\alpha}^\dagger c_{\beta k}^\dagger \right) \\ &+ c_{\alpha\beta} \left(c_{\alpha k}^\dagger c_{k\beta} + c_{\alpha k} c_{k\beta}^\dagger + c_{k\alpha}^\dagger c_{\beta k} + c_{k\alpha} c_{\beta k}^\dagger \right) \left. \right] \\ &- c_{bd} (c_{db} - 1) (c_{da} (1 - c_{ba}) + c_{dc} (1 - c_{bc})) \\ &- c_{ac} (c_{ca} - 1) (c_{da} (1 - c_{dc}) + c_{ba} (1 - c_{bc})) \\ &- c_{db} (c_{bd} - 1) (c_{bc} (1 - c_{dc}) + c_{ba} (1 - c_{da})) \\ &- c_{ca} (c_{ac} - 1) (c_{bc} (1 - c_{ba}) + c_{dc} (1 - c_{da})) \end{aligned} \quad (\text{A11})$$

3. Change in $n_{\square(\mathbf{c})}$ following one triangle-type move

The triangle move is a transformation from network \mathbf{c} to network \mathbf{x} , characterised by $x_{ij} = c_{ij} + \Omega_{ij}$ with

$$\Omega_{ij} = \delta_{ib} \delta_{ja} + \delta_{ic} \delta_{jb} + \delta_{ia} \delta_{jc} - \delta_{ia} \delta_{jb} - \delta_{ib} \delta_{jc} - \delta_{ic} \delta_{ja} \quad (\text{A12})$$

The terms which make up the square mobility term are

$$\begin{aligned} n_{\square}(\mathbf{c}) &= \frac{1}{2} \text{Tr}(\mathbf{c} \mathbf{c}^\dagger \mathbf{c} \mathbf{c}^\dagger) - \sum_{ij} k_i^{\text{out}} c_{ij} k_j^{\text{in}} + \text{Tr}(\mathbf{c} \mathbf{c}^\dagger \mathbf{c}) \\ &+ \frac{1}{2} N^2 \langle k \rangle^2 + \frac{1}{2} \text{Tr}(\mathbf{c}^2) - \sum_i k_i^{\text{out}} k_i^{\text{in}} \end{aligned}$$

- Term 2:

$$\sum_{ij} k_i^{\text{out}} \Omega_{ij} k_j^{\text{in}} = \sum_{\alpha, \beta \in \{1,2,3\}} \mathbb{I}(\alpha, \beta) k_\alpha^{\text{out}} k_\beta^{\text{in}}$$

- Term 3:

$$\begin{aligned} \Delta[\text{Tr}(\mathbf{c} \mathbf{c}^\dagger \mathbf{c})] &= \text{Tr}(\mathbf{x} \mathbf{x}^\dagger \mathbf{x}) - \text{Tr}(\mathbf{c} \mathbf{c}^\dagger \mathbf{c}) \\ &= \sum_{ijk} (c_{ij} + \Omega_{ij}) (c_{kj} + \Omega_{kj}) (c_{ki} + \Omega_{ki}) - c_{ij} c_{kj} c_{ki} \end{aligned}$$

We consider each subterm separately:

$$\begin{aligned} \sum_{ijk} \Omega_{ij} c_{kj} c_{ki} &= \sum_{ijk} \Omega_{ki} c_{kj} c_{ij} = 0 \\ \sum_{ijk} \Omega_{kj} c_{ij} c_{ki} &= \sum_{\alpha, \beta \in \{1,2,3\}} \mathbb{I}(\alpha, \beta) \sum_i c_{\alpha i} c_{i\beta} \end{aligned}$$

Clearly

$$\sum_{ijk} \Omega_{ij} \Omega_{kj} = \sum_{ijk} \Omega_{ki} \Omega_{kj} = 0$$

since the Ω kills any suffix repeated in the same position. Furthermore,

$$\sum_i \Omega_{ij} \Omega_{ki} = \sum_{\alpha, \beta \in \{a,b,c\}} (1 - \delta_{\alpha\beta}) \delta_{ja} \delta_{k\beta} - 2 \sum_{\alpha \in \{a,b,c\}} \delta_{ja} \delta_{ka}$$

hence

$$\sum_{ijk} \Omega_{ij} \Omega_{ki} c_{kj} = 3$$

So it follows that

$$\text{Tr}(\mathbf{x} \mathbf{x}^\dagger \mathbf{x}) - \text{Tr}(\mathbf{c} \mathbf{c}^\dagger \mathbf{c}) = 3 + \sum_{\alpha, \beta \in \{a,b,c\}} \mathbb{I}(\alpha, \beta) \sum_i c_{\alpha i} c_{i\beta}$$

- Term 5:

We observe that $\sum_{ij} c_{ji} \Omega_{ij} = 3$ and $\sum_{ij} \Omega_{ij} \Omega_{ji} = -6$. We conclude that $\Delta[\text{Tr}(\mathbf{c}^2)] = 0$. This is as expected, since double bonds cannot participate in a triangle swap.

- Term 1:

Finally we return to Term 1 using the various shortcuts derived above. We recall that a suffix repeated in the same position sends the term to zero. Hence, we already know that all terms featuring the product of 3 or 4 Ω terms will be zero. Next:

$$\sum_j \Omega_{ij} c_{kj} = \sum_{\alpha, \beta \in \{1,2,3\}} \mathbb{I}(\alpha, \beta) \delta_{ia} c_{k\beta}$$

From this it follows that

$$\sum_{ij} \Omega_{ij} c_{kj} \Omega_{km} c_{im} = 0$$

Finally,

$$\begin{aligned} \sum_{ijkm} \Omega_{ij} c_{kj} c_{km} c_{im} &= c_{km} [c_{k1} (c_{2m} - c_{3m}) \\ &+ c_{k2} (c_{3m} - c_{1m}) + c_{k3} (c_{1m} - c_{2m})] \end{aligned}$$

(and similarly with the other terms related to this one

by simple permutations). Overall we thus find

$$\Delta \text{Tr}(\mathbf{c}\mathbf{c}^\dagger \mathbf{c}\mathbf{c}^\dagger) = 4 \sum_{km} c_{km} \sum_{\alpha, \beta \in \{a, b, c\}} \mathbb{I}(\alpha, \beta) c_{\alpha m} c_{k\beta}$$

Collecting all these terms together, we see that the expected change in the square mobility term after the application of a single triangle type move is

$$\Delta [n_{\square}] = \sum_{\alpha, \beta \in \{a, b, c\}} \mathbb{I}(\alpha, \beta) \left[c_{\alpha i} c_{i\beta} + k_\beta^{\text{out}} k_\alpha^{\text{in}} + 2 \sum_{km} c_{km} c_{\alpha m} c_{k\beta} \right] + 3 \quad (\text{A13})$$

4. Change in $n_{\Delta}(c)$ following one triangle-type move

This final incremental term is best evaluated by an algorithm which, for each edge created or destroyed, searches for mono-directed triangles that have been created or destroyed.